

zett szerkezetű mesterséges zeolitokat is gyártanak a vegyipari technológiák különböző problémáinak megoldására.

Az ultramarinok vázában a poliéder egységek csúcsain Si és Al atomok váltakoznak, és különböző anionokat is tartalmaznak: Cl, SO₄²⁻, S₂²⁻. A di- és triszulfid ionok adják a kék színüket. Amikor semleges kénmolekulát is tartalmaznak, az ásvány színe vörös. Az ultramarinokat jelentős pigmentként használják olaj- és porcelán-festékekben, a textíliák fehéritésére alkalmazott kékítő készítésére.

Kultúrtörténeti érdekesség, hogy W. Goethe német költő már az 1780-as évek végén egy közleményében javasolta az ultramarin mesterséges előállítását. Ötletét Gmelin, neves vegyész valósította meg, amit a méissenai porcelángyárban ipari méretekben is alkalmaztak.

Forrásanyag:

N.N.Greenwood, A.Earnshau: *Az elemek kémiája*, Nemzeti Tk., Bp. 1999.

Katona Imre: *A porcelán, Gondolat Zsebkönyvek, Bp. 1984*

Máthé Enikő



Négyzetes mátrixok forgatása

Feladat

Forgassunk el egy $n \times n$ -es mátrixot az óramutató járásával ellenkező irányba!

1. nem hatékony megoldás

A tapasztalatlanabb diákok intuitíven, első látásra úgy oldják meg a feladatot, hogy vesznek egy segédmátrixot, abba átmásolják átforgatva a mátrixot. Természetesen az átforgatáshoz az indexek kiszámítása jelenthet problémát.

Elemzés, megbeszélés

– Hogyan forgathatjuk át a mátrixot?

Például a következő mátrix elforgatva így néz ki:

1	2	3
4	5	6
7	8	9

→

3	6	9
2	5	8
1	4	7

vagy indexekkel felírva:

0, 0	0, 1	0, 2
1, 0	1, 1	1, 2
2, 0	2, 1	2, 2

→

0, 2	1, 2	2, 2
0, 1	1, 1	2, 1
0, 0	1, 0	2, 0

Azonnal megfigyelhetjük, hogy ami az eredeti mátrixban az i index értéke volt, az most a mátrix oszlopaiban veszi fel szerepét, mégpedig fordított sorrendben, és hason-

lón, ami az eredeti mátrixban a j index értéke volt, az most a mátrix soraiban veszi fel szerepét.

A megoldás tehát a következő:

```
#include<stdio.h>

int main()
{
    int n, i, j;
    int t[20][20];
    // n beolvasása
    printf("n: ");
    scanf("%i", &n);
    // a mátrix beolvasása
    for(i=0; i<n; ++i)
        for(j=0; j<n; ++j)
        {
            printf("t[%i][%i]: ", i, j);
            scanf("%i", &t[i][j]);
        }
    // a mátrix kiírása
    for(i=0; i<n; ++i)
    {
        for(j=0; j<n; ++j)
            printf("%4i", t[i][j]);
        printf("\n");
    }
    printf("\n");
    // a mátrix átforgatása
    int b[20][20];
    for(i=0; i<n; ++i)
        for(j=0; j<n; ++j)
            b[i][j]=t[j][n-i-1];
    // a mátrix átmásolása
    for(i=0; i<n; ++i)
        for(j=0; j<n; ++j)
            t[i][j]=b[i][j];
    // a mátrix kiírása
    for(i=0; i<n; ++i)
    {
        for(j=0; j<n; ++j)
            printf("%4i", t[i][j]);
        printf("\n");
    }
    return 0;
}
```

Elemzés, megbeszélés

- A fenti megoldás nem hatékony, ugyanakkora mátrixot használ fel segédadatként az elforgatáshoz. Hogyan tehető hatékonyvá?
- Elemezzük két változó értékének a felcserélését:
 $a \leftrightarrow b$ azt jelenti, hogy:

```
int c;
```

c=a;
a=b;
b=a;

Egy segédváltozó segítségével felcseréltük az a és a b változók értékeit.

– Próbáljuk ezt most megoldani segédváltozó használata nélkül!

Matematikailag könnyen igazolható, hogy két változó (a és b) értékét felcserélhetjük egymással segédváltozók használata nélkül, a következő utasítássorozattal:

a=a+b;
b=a-b;
a=a-b;

Vigyázzunk azonban, mert ez csak akkor működik, ha a műveletek elvégzésekor nem lép fel túlsordulás!

– Igen ám, de nekünk a mátrix elforgatásához négy változó értékének a körkörös felcserélésére van szükségünk. Ez is megoldható!

Három változó esetén az utasítássorozat így alakul:

a=a+b+c;
b=a-b-c;
c=a-b-c;
a=a-b-c;

Négy változó esetén pedig így:

a=a+b+c+d;
b=a-b-c-d;
c=a-b-c-d;
d=a-b-c-d;
a=a-b-c-d;

– A segédváltozók tehát kiküszöbölhetők! Nézzük most meg, hogy milyen elemeket kell elforgatni:

Például a fenti mátrix elforgatva így néz ki:

1	2	3	→	3	6	9
4	5	6		2	5	8
7	8	9		1	4	7

Vagyis a következő cseréket hajtottuk végre:

t[0][0] ← t[2][0] ← t[2][2] ← t[0][2] ← t[0][0]
t[0][1] ← t[1][0] ← t[2][1] ← t[1][2] ← t[0][1]

Tehát a mátrix következő elemei a kiindulási pontok a cserékhez:

1	2	3
4	5	6
7	8	9

4×4-es mátrix esetén már a következő elemek lesznek a kiindulási pontok a négyes cserékhez:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

5×5-ös mátrix esetén pedig:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

6×6-os mátrix esetén pedig:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Észrevehetjük, hogy a ciklusok csak a szürkével jelölt háromszöget kell, hogy bejárják, és minden elemre ebből a háromszögből végre kell, hogy hajtsák a neki megfelelő elemek körkörös cseréjét.

Megoldandó probléma tehát a ciklusok felírása (az indexek határainak meghatározása), valamint a megfelelő elemek indexeinek kiszámítása.

Az indexhatárok meghatározásánál a fenti példákban egyértelműen kitűnik, hogy az i sorindex csak a mátrix felét kell, hogy bejárja, vagyis az első ciklus így alakul:

```
for (i=0; i<n/2; ++i)
```

A j oszlopindex határainál pedig észrevesszük, hogy az első sorban a j 0-tól $n-1$ -ig megy, a második sorban 1-től $n-2$ -ig, a harmadik sorban 2-től $n-3$ -ig és így tovább. A j határai tehát kifejezhetők i és n függvényében:

```
for (j=i; j<n-i-1; ++j)
```

Az indexhatárokat letisztáztuk, nem marad más hátra, mint a cseréhez szükséges elemek indexeinek a meghatározása. Induljunk ki ismét a 3×3 -as mátrixból:

1	2	3
4	5	6
7	8	9

Vagyis a következő cseréket hajtottuk végre:

```
t[0][0] ← t[2][0] ← t[2][2] ← t[0][2] ← t[0][0]
t[0][1] ← t[1][0] ← t[2][1] ← t[1][2] ← t[0][1]
```

A 4×4 -es mátrix esetén pedig:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Vagyis a következő cseréket hajtottuk végre:

```
t[0][0] ← t[3][0] ← t[3][3] ← t[0][3] ← t[0][0]
t[0][1] ← t[2][0] ← t[3][2] ← t[1][3] ← t[0][1]
t[0][2] ← t[1][0] ← t[3][1] ← t[2][3] ← t[0][2]
t[1][1] ← t[2][1] ← t[2][2] ← t[1][2] ← t[1][1]
```

A cserék kiindulópontjai mindig a $t[i][j]$ elemek, általánosan pedig így írható le a körkörös csere (a fenti és az első megoldásból származó gondolatmenet alapján):

```
t[i][j] ← t[n-j-1][i] ← t[n-i-1][n-j-1] ← t[j][n-i-1]
```

A feladat hatékony megoldása tehát a következő:

2. megoldás

```
for (i=0; i<n/2; ++i)
    for (j=i; j<n-i-1; ++j)
```

```

{
  t[i][j]=t[i][j]+t[n-j-1][i]+t[n-i-1][n-j-1]+t[j][n-i-1];
  t[n-j-1][i]=t[i][j]-t[n-j-1][i]-t[n-i-1][n-j-1]-t[j][n-i-1];
  t[n-i-1][n-j-1]=t[i][j]-t[n-j-1][i]-t[n-i-1][n-j-1]-t[j][n-i-1];
  t[j][n-i-1]=t[i][j]-t[n-j-1][i]-t[n-i-1][n-j-1]-t[j][n-i-1];
  t[i][j]=t[i][j]-t[n-j-1][i]-t[n-i-1][n-j-1]-t[j][n-i-1];
}

```

Így segédváltozókat sem használtunk és a ciklusok is pontosan annyi elemet járnak be, amennyi feltétlenül szükséges.

Megjegyzések

Két változó értéke felcserélhető az xor (kizáró vagy) művelet segítségével is. C-ben, C++-ban az xor műveletet a ^ jelöli. A művelet táblája:

a	b	a^b
0	0	0
0	1	1
1	0	1
1	1	0

A cseréhez szükséges utasítássorozat pedig a következő:

```

b=a^b;
a=a^b;
b=a^b;

```

Kovács Lehel

A labdarúgás fizikája

I. rész

Bevezetés

A labdarúgás (futball – angol eredetű szóval) egy labdajáték, amelyet a futballpályán (1. ábra) két egyenként 11 labdarúgóból álló csapat játszik egymás ellen. A játék célja: a játékidő alatt az ellenfél kapujába jutassák a labdát (gólt szerezzenek). A játékot főleg lábbal játsszák, de a játékos minden testrészét használhatja a labda irányítására a két karján kívül. A 16-os vonalon belül a kapusok a kezüket is használhatják. A játékszabályok betartására a futballbíró vigyáz. A labda gömb alakú, legnagyobb gömbi körének a kerülete $68 \div 70$ cm. A labda tömege $0,41 \div 0,45$ kg és a benne levő levegő túlnyomása a meccs kezdetekor $0,6 \div 1,1$ bar. A Nemzetközi Labdarúgószövetség (FIFA) becslései szerint világszerte több mint 240 millió ember játssza rendszeresen 208 országban. A legtöbb országban ez a legnépszerűbb sport.

Miközben dobjuk, rúgjuk, fejeljük a labdát, nemigen gondolunk arra, hogy a sikeres játék feltétele az, hogy megfelelően alkalmazkodunk a labda mozgását megszabó alapvető fizikai törvényekhez. A következőkben a labdajátékkal kapcsolatban felvetődő néhány kérdés (mekkora lehet a labda sebessége, milyen alakú a röppálya, mekkora egy labda erőhatása ütközéskor, mennyi az ütközési idő, hogyan változik meg a labda sebessége ferde ütközéskor) megválaszolásával foglalkozunk a fizika törvényei alapján.